SIEMENS

SIMATIC

S7ProSim V5.4

Objeto COM

Manual del usuario

Edición: 01/2007

Copyright y consignas de seguridad

Este manual contiene las informaciones necesarias para la seguridad personal, así como para la prevención de daños materiales. Las señales que figuran a continuación representan distintos grados de peligro:

Peligro
Significa que, si no se adoptan las medidas preventivas adecuadas, se producirá la muerte, o bien lesiones corporales graves o daños materiales considerables.

Precaución
Significa que, si no se adoptan las medidas preventivas adecuadas, puede producirse la muerte, lesiones corporales graves o daños materiales considerables.

Cuidado
Junto con el símbolo de aviso, significa que, si no se adoptan las medidas preventivas adecuadas, pueden producirse lesiones corporales leves o moderadas.

Cuidado
Sin el símbolo de aviso, significa que, si no se adoptan las medidas preventivas adecuadas, pueden producirse daños materiales.

Nota Sin el símbolo de aviso, significa que, si no se adoptan las medidas preventivas adecuadas, pueden producirse resultados o estados impredecibles.

Personal cualificado

La puesta en funcionamiento y el servicio del equipo sólo deben ser llevados a cabo conforme a lo descrito en este manual. Sólo está autorizado a intervenir en este equipo el personal cualificado. En el sentido del manual se trata de personas que disponen de los conocimientos técnicos necesarios para poner en funcionamiento, conectar a tierra y marcar los aparatos, sistemas y circuitos de acuerdo con las normas estándar de seguridad.

Uso conforme

Considere lo siguiente:

Precaución
El equipo y los componentes del sistema sólo se podrán utilizar para los casos de aplicación previstos en el catálogo y en la descripción técnica, y sólo con los equipos y componentes de proveniencia tercera recomendados y homologados por Siemens.

El funcionamiento correcto y seguro del producto presupone un transporte, un almacenamiento, una instalación y un montaje conforme a las prácticas de la buena ingeniería, así como un manejo y un mantenimiento rigurosos.

Marcas registradas

Siemens® y SIMATIC® son marcas registradas de SIEMENS AG.
STEP 7™ y S7™ son marcas registradas de SIEMENS AG.
Microsoft ®, Windows ®, Windows 95 ®, Windows 98 ®, Windows NT ®, Windows ME ® y Windows 2000 ® son marcas registradas de Microsoft Corporation.

Copyright Siemens AG, 2007
Reservados todos los derechos
La divulgación y reproducción de este documento, así como el uso y la comunicación de su contenido, no están autorizados, a no ser que se obtenga el consentimiento expreso para ello. Los infractores quedan obligados a la indemnización de los daños. Se reservan todos los derechos, en particular para el caso de concesión de patentes o de modelos de utilidad.

Exención de responsabilidad
Hemos probado el contenido de esta publicación con la concordancia descrita para el hardware y el software. Sin embargo, es posible que se den algunas desviaciones que nos impiden tomar garantía completa de esta concordancia. El contenido de esta publicación está sometido a revisiones regularmente y en caso necesario se incluyen las correcciones en la siguiente edición. Agradecemos sugerencias.

Siemens AG
Automation and Drives
Postfach 4848, D-90327 Nuernberg

© Siemens AG 2007

Sujeto a cambios sin previo aviso.

Prólogo

S7ProSim ofrece un acceso programático a la interfaz de simulación de procesos de S7-PLCSIM. S7ProSim permite escribir programas para realizar tareas tales como cambiar la posición del selector de modo del PLC simulado, ejecutar un solo ciclo del programa de control, leer o escribir valores de/en el PLC y muchas otras funciones más.

Destinatarios

El manual está dirigido a ingenieros, programadores y técnicos de instalación que dispongan de conocimientos sobre los sistemas de automatización S7 y que tengan experiencia en la programación con Visual Basic (6.0 o .NET), o bien Visual C++ (6.0 o .NET).

Objeto del manual

Este documento describe las características y el funcionamiento de S7ProSim V5.4.

Manuales complementarios

Encontrará más información en la Ayuda en pantalla de STEP 7 y S7-PLCSIM, así como en los manuales indicados a continuación:

- *Programar con STEP 7*. Este manual proporciona informaciones básicas sobre cómo diseñar y crear programas de control. Utilice este manual si desea crear un programa de control con el software de automatización STEP 7.
- *Software de sistema para SIMATIC S7-300/400 – Funciones estándar y funciones de sistema*. En este manual se describen las funciones del sistema, los bloques de organización y las funciones estándar para desarrollar un programa de control.
- *STEP 7 – Introducción y ejercicios prácticos – Getting Started*. Este manual explica cómo utilizar el software de automatización STEP 7, ofreciendo una panorámica de los procedimientos utilizados para configurar un PLC y para desarrollar programas de control.
- *S7-PLCSIM – Comprobar el programa de la CPU S7.* En este manual se describe la interfaz de usuario y el funcionamiento de S7-PLCSIM, el simulador de PLCs S7.

Para encontrar estos y otros manuales más, elija los comandos **Inicio > Simatic > Documentación** a partir del menú Inicio del PC en el que está instalado STEP 7.

Asistencia técnica

Si tiene preguntas técnicas, si necesita información sobre los cursillos de entrenamiento en relación con este producto, o bien para efectuar pedidos, diríjase por favor a su representante de Siemens más próximo..

Norteamérica y Suramérica	Europa y África	Asia y región del Pacífico
Teléfono: +1 (800) 333-7421	Teléfono: +49 (0) 180 5050 222	Teléfono: +86 10 64 75 75 75
Fax: +1 (423) 262-2200	Fax: +49 (0) 180 5050 223	Fax: +86 10 64 74 74 74
simatic.hotline@siemens.com	adsupport@siemens.com	adsupport.asia@siemens.com

Contenido

Resumen breve de S7ProSim 7

Agregar un objeto COM S7ProSim al proyecto 7
Utilizar S7ProSim para programar una interfaz con S7-PLCSIM 8

Métodos 11

BeginScanNotify 13
Connect 14
Continue 15
Disconnect 16
EndScanNotify 17
ExecuteNmsScan 18
ExecuteNScans 19
ExecuteSingleScan 20
GetPauseState 21
GetScanMode 22
GetStartUpSwitch 23
GetState 24
HotStartWithSavedValues 25
Pause 26
ReadDataBlockValue 27
ReadFlagValue 28
ReadOutputImage 29
ReadOutputPoint 30
SavePLC 31
SetScanMode 32
SetStartUpSwitch 33
SetState 34
StartPLCSim 35
WriteDataBlockValue 36
WriteFlagValue 37
WriteInputImage 38
WriteInputPoint 39

Eventos 41

ConnectionError 42
PauseStateChanged 43
PLCSimStateChanged 44
ScanFinished 45
ScanModeChanged 46

Definiciones de tipo 47
CPURunMode 48
ImageDataTypeConstants 49
PauseStateConstants 50
PointDataTypeConstants 51
RestartSwitchPosition 52
ScanModeConstants 53
tagPauseState 54
ScanInfo Constants 55
Códigos de error 59
Índice 61

Resumen breve de S7ProSim

S7ProSim es un objeto COM que ofrece un acceso programático a la interfaz de simulación de procesos de S7-PLCSIM. S7ProSim puede utilizarse en cualquier aplicación que acepte objetos COM para establecer una conexión con una simulación de procesos de S7-PLCSIM.

Este documento electrónico describe cómo agregar S7ProSim a una aplicación, así como las funciones, la interfaz y las operaciones de S7ProSim, incluyendo definiciones de objetos de software de los métodos y eventos.

Agregar un objeto COM S7ProSim al proyecto

Para utilizar un objeto COM S7ProSim en el proyecto es preciso agregar una referencia a dicho objeto. Los pasos necesarios para agregar una referencia a un objeto COM de S7ProSim dependen del entorno de programación. Por ejemplo, en Microsoft Visual Basic (6.0 o .NET), deberá seguir los pasos indicados a continuación:

1. Elija el comando de menú **Proyecto > Referencias** o **Proyecto > Agregar referencia**.
2. En el cuadro de diálogo "Referencias", active la casilla de verificación correspondiente a "Siemens S7ProSim COM Object". (En Visual Basic .NET, está selección se efectúa en la ficha "COM" del cuadro de diálogo "Referencias").
3. Haga clic en "Aceptar".

Tras haber agregado la referencia al proyecto, puede utilizar el Examinador de objetos para examinar los métodos y eventos del objeto COM S7ProSim. En el Examinador de objetos, seleccione "S7PROSIMLib" en la lista desplegable de bibliotecas. La clase S7ProSim contiene los métodos y eventos que pueden utilizarse para programar una interfaz con S7-PLCSIM.

En Microsoft Visual Studio C++ V6.0 o en Microsoft Visual C++ .NET, siga los procedimientos necesarios en ese entorno de programación para agregar un objeto COM.

Utilizar S7ProSim para programar una interfaz con S7-PLCSIM

Para utilizar S7ProSim con objeto de programar funciones en el PLC simulado en S7-PLCSIM deberá realizar las tareas siguientes:

- Incorporar el objeto COM S7ProSim de Siemens en el proyecto.
- Agregar una referencia del proyecto para S7ProSim.

  Ejemplo: Visual Basic 6.0

```
Option Explicit
Private WithEvents S7ProSim As S7PROSIMLib.S7ProSim
...
Private Sub Form_Load()
Set S7ProSim = New S7PROSIMLIB.S7ProSim
...
End Sub
```

  Ejemplo: Visual Basic .NET

```
Private WithEvents S7ProSim As New S7PROSIMLib.S7ProSim
```

  Ejemplo: Visual C++ 6.0

```
// la ProSim library/tlb está en el archivo dll

#import <S7wspsmx.dll> named_guids, no_namespace//, raw_interfaces_only

class ProSimWrapper
{
public:
    ProSimWrapper() : m_pProSim(OLESTR("S7wspsmx.S7ProSim"), NULL,
CLSCTX_INPROC_SERVER)
    {};    // el spartptr se crea automáticamente en
           // la pila cuando se inicia la aplicación

    virtual ~ProSimWrapper()
    {};    // no implementado, el smartptr se libera
           // automáticamente cuando se cierra la aplicación

    IS7ProSim * GetPtr()
    {
        return m_pProSim;
    };

// Atributos
protected:
    // IProSimPtr es un CComPtr (smart ptr) de la interfaz IProSim
    // Es del archivo dll del #import
    // CoCreateInstance se llama automáticamente en el objeto
    // ptr en el constructor de esta clase liberar ptr
    // es llamado automáticamente por el destructor de esta clase
    IS7ProSimPtr    m_pProSim;
};
```

  Ejemplo: C#

```
using S7PROSIMLib;
...
private S7ProSim ps;
```

- En Visual Basic, programar procesadores de los eventos de S7ProSim. Los procesadores de eventos no son indispensables en Visual C++. En cada controlador de eventos es posible insertar cualquier código personalizado para la aplicación.

Ejemplo: Visual Basic 6.0

```
Private Sub S7ProSim_PauseStateChanged(ByVal NewState As String)
DoEvents
    ...
End Sub

Private Sub S7ProSim_ScanFinished(ByVal ScanInfo As Variant)
DoEvents
    ...
End Sub

Private Sub S7ProSim_PLCSimStateChanged(ByVal NewState As String)
    DoEvents
    ...
End Sub

Private Sub S7ProSim_ConnectionError(ByVal ControlEngine As String, ByVal
error As Long)
    DoEvents
    MsgBox "Connection Error"
End Sub

Private Sub S7ProSim_ScanModeChanged(ByVal NewState As String)
DoEvents
    ...
End Sub
```

Nota
La llamada a "`DoEvents`" no es necesaria en Visual Basic .NET.

- Agregue botones de comando, cuadros de texto u otros objetos a su aplicación según sea necesario para acceder a los diversos métodos de S7ProSim. Programe el código para cada controlador de botones de comando, a fin de llamar a los métodos de S7ProSim y ajustar los valores correspondientes para los cuadros de texto deseados.

Métodos

BeginScanNotify	Registra a S7ProSim para devoluciones de llamadas del PLC. Los eventos ScanFinished y PLCSimStateChanged se enviarán cuando éstos ocurran.
Connect	Conecta S7ProSim con S7-PLCSIM.
Continue	Continúa una simulación que ha sido detenida.
Disconnect	Desconecta S7ProSim de S7-PLCSIM.
EndScanNotify	Elimina S7ProSim del registro para devoluciones de llamadas del PLC. Los eventos ScanFinished y PLCSimStateChanged no se envían.
ExecuteNmsScan	Obliga a S7-PLCSIM a ejecutar ciclos durante un tiempo determinado (Nms), sin esperar a que culmine la ejecución del ciclo actual. Si se habilita la notificación, el programa recibirá un aviso cuando S7-PLCSIM haya finalizado los ciclos.
ExecuteNScans	Obliga a S7-PLCSIM a ejecutar un número determinado de ciclos sin esperar a que culmine la ejecución del ciclo actual. Si se habilita la notificación, el programa recibirá un aviso cuando S7-PLCSIM haya finalizado los ciclos.
ExecuteSingleScan	Obliga a S7-PLCSIM a ejecutar un ciclo individual sin esperar a que culmine la ejecución del ciclo actual. Si se habilita la notificación, el programa recibirá un aviso cuando S7-PLCSIM haya finalizado el ciclo.
GetPauseState	Devuelve el estado de pausa actual de S7-PLCSIM.
GetScanMode	Devuelve el modo de ciclo de S7-PLCSIM.
GetStartUpSwitch	Obtiene la configuración de arranque (rearranque, rearranque completo o arranque en frío) de S7-PLCSIM.
GetState	Devuelve una cadena que contiene la posición actual del selector de modo de S7-PLCSIM (RUN, RUN-P o STOP).
HotStartWithSavedValues	Activa un valor booleano para determinar si S7-PLCSIM debe cargar datos de E/S de periferia cuando se inicie en el estado HotStart. Para que S7-PLCSIM pueda arrancar y cargar E/S de periferia, el usuario deberá llamar a HotStartWithSavedValues con el valor TRUE, guardar el programa del PLC (SavePLC), y ajustar el estado de arranque de S7-PLCSIM HotStart (SetStartUpSwitch). S7-PLCSIM cargará las E/S de periferia durante el arranque.
Pause	Detiene una simulación.
ReadDataBlockValue	Lee un bit, un byte, una palabra o una palabra doble del área de memoria DB de S7-PLCSIM.
ReadFlagValue	Lee un bit, un byte, una palabra o una palabra doble del área de memoria de marcas (M) de S7-PLCSIM.
ReadOutputImage	Lee elementos de la imagen de salidas de periferia (área de memoria PQ) de S7-PLCSIM.
ReadOutputPoint	Lee un bit en particular (Boolean), un byte (Byte), una palabra de 16 bits (Integer) o una palabra de 32 bits (Long) del área de las salidas periféricas (PQ).

SavePLC	Guarda en un archivo los datos actuales del PLC simulado. Los datos que se almacenan incluyen el programa, la configuración de hardware, la posición del selector de modo que indica la subventana "CPU", el tipo de ciclo (continuo o individual), el estado de E/S, los valores de los temporizadores, las direcciones simbólicas y el estado de la alimentación ("on" u "off").
SetScanMode	Ajusta el modo de ciclo de S7-PLCSIM.
SetStartUpSwitch	Ajusta el tipo de arranque (rearranque, rearranque completo o arranque en frío) de S7-PLCSIM.
SetState	Ajusta la posición actual del selector de modo de S7-PLCSIM (RUN, RUN-P o STOP).
StartPLCSim	Inicia S7-PLCSIM con el archivo indicado de simulación del PLC (guardado en una llamada anterior a SavePLC).
WriteDataBlockValue	Escribe un bit, un byte, una palabra o una palabra doble en el área de memoria DB de S7-PLCSIM.
WriteFlagValue	Escribe un bit, un byte, una palabra o una palabra doble en el área de memoria de marcas (M) de S7-PLCSIM.
WriteInputImage	Escribe elementos en la imagen de entradas de periferia (área de memoria PI) de S7-PLCSIM, comenzando en el StartIndex de los datos a los que señala pData.
WriteInputPoint	Escribe un bit en particular (Boolean), un byte (Byte), una palabra de 16 bits (Integer) o una palabra de 32 bits (Long) del Data Variant en el área de las entradas de periferia (PI).

BeginScanNotify

STDMETHOD(**CS7ProSim::BeginScanNotify**)()

Descripción

Registra a S7ProSim para devoluciones de llamadas del PLC. Los eventos ScanFinished y PLCSimStateChanged se enviarán cuando éstos ocurran.

Parámetros

Ninguno

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado

Uso en Visual Basic

```
Function BeginScanNotify() As Long
```

Connect

`STDMETHOD(CS7ProSim::Connect)()`

Descripción

Conecta S7ProSim con S7-PLCSIM.

Parámetros

Ninguno

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado

Uso en Visual Basic

```
Function Connect() As Long
```

Continue

STDMETHOD(**CS7ProSim::Continue**)()

Descripción

Continúa una simulación que ha sido detenida.

Parámetros

Ninguno

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Sub Continue()
```

Disconnect

`STDMETHOD(CS7ProSim::Disconnect)()`

Descripción

Desconecta S7ProSim de S7-PLCSIM.

Parámetros

Ninguno

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado

Uso en Visual Basic

```
Function Disconnect() As Long
```

EndScanNotify

`STDMETHOD(CS7ProSim::EndScanNotify)()`

Descripción

Elimina S7ProSim del registro para devoluciones de llamadas del PLC. Los eventos ScanFinished y PLCSimStateChanged no se envían.

Parámetros

Ninguno

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado
PS_E_NOTREGISTERED	0x80040209 : S7ProSim no está registrado para devoluciones de llamadas de S7-PLCSIM

Uso en Visual Basic

```
Function EndScanNotify() As Long
```

ExecuteNmsScan

STDMETHOD(**CS7ProSim::ExecuteNmsScan**)(long *MsNumber*)

Descripción

Obliga a S7-PLCSIM a ejecutar ciclos durante un tiempo determinado (Nms), sin esperar a que culmine la ejecución del ciclo actual. Si se habilita la notificación, el programa recibirá un aviso cuando S7-PLCSIM haya finalizado los ciclos. Para poder utilizar este método, S7-PLCSIM deberá estar en modo de ciclo individual.

Parámetros

MsNumber Tiempo (en milisegundos) durante el que deben ejecutarse los ciclos del programa.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_NOTSINGLESCAN	0x8004020A : S7-PLCSIM no está en modo de ciclo individual
PS_E_PLCNOTRUNNING	0x8004020E : S7-PLCSIM no se está ejecutando
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

Function **ExecuteNmsScan**(*MsNumber As Long*) As Long

ExecuteNScans

STDMETHOD(**CS7ProSim::ExecuteNScans**)(long *NScanNumber*)

Descripción

Obliga a S7-PLCSIM a ejecutar un número determinado de ciclos sin esperar a que culmine la ejecución del ciclo actual. Si se habilita la notificación, el programa recibirá un aviso cuando S7-PLCSIM haya finalizado los ciclos. Para poder utilizar este método, S7-PLCSIM deberá estar en modo de ciclo individual.

Parámetros

NScanNumber Número de ciclos a ejecutar

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_NOTSINGLESCAN	0x8004020A : S7-PLCSIM no está en modo de ciclo individual
PS_E_PLCNOTRUNNING	0x8004020E : S7-PLCSIM no se está ejecutando
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

Function **ExecuteNScans**(*NScanNumber As Long*) As Long

ExecuteSingleScan

STDMETHOD(**CS7ProSim::ExecuteSingleScan**)()

Descripción

Obliga a S7-PLCSIM a ejecutar un ciclo individual sin esperar a que culmine la ejecución del ciclo actual. Si se habilita la notificación, el programa recibirá un aviso cuando S7-PLCSIM haya finalizado el ciclo. Para poder utilizar este método, S7-PLCSIM deberá estar en modo de ciclo individual.

Parámetros

Ninguno

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_PLCNOTRUNNING	0x8004020E : S7-PLCSIM no se está ejecutando
PS_E_NOTSINGLESCAN	0x8004020A : S7-PLCSIM no está en modo de ciclo individual
PS_E_MODENOTPOSSIBLE	0x8004020C : S7-PLCSIM no ha podido activar el modo de ciclo indicado

Uso en Visual Basic

```
Function ExecuteSingleScan() As Long
```

GetPauseState

```
STDMETHOD(CS7ProSim::GetPauseState)( PauseStateConstants *pVal)
```

Descripción

Devuelve el estado de pausa actual de S7-PLCSIM.

Parámetros

pVal Puntero al estado devuelto de S7-PLCSIM que es una de las PauseStateConstants

Notas

Si se llama desde Visual Basic, el estado de pausa se devuelve en el valor de retorno de la función y no hay *pVal* parameter.

Si se llama desde C++, el estado se devuelve en el valor de retorno al que señala *pVal*.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Function GetPauseState() As PauseStateConstants
```

GetScanMode

```
STDMETHOD(CS7ProSim::GetScanMode)( ScanModeConstants *pVal)
```

Descripción

Devuelve el modo de ciclo de S7-PLCSIM.

Parámetros

pVal	Puntero al modo de ciclo devuelto. El modo de ciclo devuelto es una de las ScanModeConstants

Notas

Si se llama desde Visual Basic, el modo de ciclo se devuelve en el valor de retorno de la función y no hay *pVal* parameter.

Si se llama desde C++, el estado se devuelve en el valor de retorno al que señala *pVal*.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Function GetScanMode() As ScanModeConstants
```

GetStartUpSwitch

```
STDMETHOD(CS7ProSim::GetStartUpSwitch)( RestartSwitchPosition *pPos)
```

Descripción

Obtiene la configuración de arranque (rearranque, rearranque completo o arranque en frío) de S7-PLCSIM.

Parámetros

pPos	Puntero al valor de posición de arranque de S7-PLCSIM que es una de las RestartSwitchPosition

Notas

Si se llama desde Visual Basic, la posición del interruptor de arranque se devuelve en el valor de retorno de la función y no hay *pPos*.

Si se llama desde C++, el estado se devuelve en el valor de retorno al que señala *pPos*.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Function GetStartUpSwitch() As RestartSwitchPosition
```

GetState

```
STDMETHOD(CS7ProSim::GetState)( BSTR *pVal)
```

Descripción

Devuelve una cadena que contiene la posición actual del selector de modo de S7-PLCSIM (RUN, RUN-P o STOP).

Parámetros

pVal Puntero al valor devuelto de la posición del selector de modo de S7-PLCSIM.

Notas

Si se llama desde Visual Basic, el estado se devuelve en el valor de retorno de la función y no hay *pVal* parameter.

Si se llama desde C++, el estado se devuelve en el valor de retorno al que señala *pVal*.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
E_INVALID_STATE	0x00008002 : Estado no válido
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Function GetState() As String
```

HotStartWithSavedValues

STDMETHOD(**CS7ProSim::HotStartWithSavedValues**)(BOOL *val*)

Descripción

Activa un valor booleano para determinar si S7-PLCSIM debe cargar datos de E/S de periferia cuando se inicie en el estado HotStart.

Para que S7-PLCSIM pueda arrancar y cargar E/S de periferia, el usuario deberá llamar a **HotStartWithSavedValues** con el valor TRUE, guardar el programa del PLC (SavePLC) y ajustar el estado de arranque de S7-PLCSIM HotStart (SetStartUpSwitch). S7-PLCSIM cargará las E/S de periferia durante el arranque.

Parámetros

val	El valor TRUE indica que S7-PLCSIM debe cargar datos de las E/S de periferia durante el rearranque. El valor FALSE indica que no debe hacerlo.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Sub HotStartWithSavedValues(val As Long)
```

Pause

```
STDMETHOD(CS7ProSim::Pause)()
```

Descripción

Detiene una simulación.

Parámetros

Ninguno

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Sub Pause()
```

ReadDataBlockValue

```
STDMETHOD(CS7ProSim::ReadDataBlockValue)(
                long BlockNumber,
                long ByteIndex,
                long BitIndex,
                PointDataTypeConstants DataType,
                VARIANT* pData)
```

Descripción

Lee un bit, un byte, una palabra o una palabra doble del área de memoria DB de S7-PLCSIM.

Parámetros

BlockNumber	Número del bloque de datos a leer. Los valores válidos de *BlockNumber* dependen de la CPU.
ByteIndex	Posición del byte inicial en el bloque de datos a leer. Los valores válidos de *ByteIndex* dependen de la CPU.
BitIndex	Posición del bit inicial en el bloque de datos a leer (si se lee un valor (de bit) booleano). Los valores válidos de *BitIndex* están comprendidos entre 0 y 7.
DataType	Tipo de datos a leer. *DataType* debe ser una de las PointDataTypeConstants.
pData	Puntero al espacio del valor devuelto. Es preciso asignar y liberar esta área de memoria en la aplicación de usuario.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado
PS_E_BADTYPE	0x80040206 : Tipo de datos no válido
PS_E_BADBYTENDX	0x80040201 : Índice de byte no válido
PS_E_BADBYTECOUNT	0x80040202 : El tamaño del array de datos no es válido para el índice del byte inicial indicado
PS_E_READFAILED	0x80040203 : Ha fallado la operación de lectura

Uso en Visual Basic

```
Sub ReadDataBlockValue(BlockNum As Long, ByteIndex As Long, BitIndex As Long, DataType As PointDataTypeConstants, pData)
```

ReadFlagValue

```
STDMETHOD(CS7ProSim::ReadFlagValue)( long ByteIndex,
                                     long BitIndex,
                                     PointDataTypeConstants DataType,
                                     VARIANT* pData)
```

Descripción

Lee un bit, un byte, una palabra o una palabra doble del área de memoria de marcas (M) de S7-PLCSIM.

Parámetros

ByteIndex	Representa la posición del byte inicial en la memoria M a leer. Los valores válidos de *ByteIndex* dependen de la CPU.
BitIndex	Representa la posición del byte inicial en la memoria M a leer (si se lee un valor (de bit) booleano). Los valores válidos de *BitIndex* están comprendidos entre 0 y 7.
DataType	Representa el tipo de datos a leer. *DataType* debe ser una de las PointDataTypeConstants.
pData	Puntero al espacio del valor devuelto. Es preciso asignar y liberar esta área de memoria en la aplicación de usuario.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado
PS_E_BADTYPE	0x80040206 : Tipo de datos no válido
PS_E_BADBYTENDX	0x80040201 : Índice de byte no válido
PS_E_BADBYTECOUNT	0x80040202 : El tamaño del array de datos no es válido para el índice del byte inicial indicado
PS_E_READFAILED	0x80040203 : Ha fallado la operación de lectura

Uso en Visual Basic

```
Sub ReadFlagValue(ByteIndex As Long, BitIndex As Long, DataType As PointDataTypeConstants, pData)
```

ReadOutputImage

```
STDMETHOD(CS7ProSim::ReadOutputImage)( long StartIndex,
                                       long ElementsToRead,
                                       ImageDataTypeConstants DataType,
                                       VARIANT* pData)
```

Descripción

Lee elementos de la imagen de salidas de periferia (área de memoria PQ) de S7-PLCSIM.

Parámetros

StartIndex	Representa la posición del byte inicial en el búfer de imagen de las salidas de periferia a leer. Los valores válidos de *StartIndex* dependen de la CPU.
ElementsToRead	Representa el número de bytes, palabras o palabras dobles a leer del búfer de imagen. Los valores válidos de *ElementsToRead* dependen de la CPU.
DataType	Representa el tipo de datos a leer. El valor de *DataType* debe ser una de las ImageDataTypeConstants.
pData	Puntero al espacio de los elementos devueltos. Los valores válidos de los datos dependen de *ElementsToRead*. Es preciso asignar y liberar esta área de memoria en la aplicación de usuario.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_BADBYTENDX	0x80040201 : Índice de byte no válido
PS_E_BADBYTECOUNT	0x80040202 : El tamaño del array de datos no es válido para el índice del byte inicial indicado
PS_E_READFAILED	0x80040203 : Ha fallado la operación de lectura
PS_E_BADTYPE	0x80040206 : Tipo de datos no válido
PS_E_NOTALLREADSWORKED	0x8004020F : Han fallado todas las operaciones de lectura
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado

Uso en Visual Basic

```
Function ReadOuputImage(StartIndex As Long, ElementsToRead As Long, DataType As ImageDataTypeConstants, pData) As Long
```

ReadOutputPoint

```
STDMETHOD(CS7ProSim::ReadOutputPoint)( long ByteIndex,
                                       long BitIndex,
                                       PointDataTypeConstants DataType,
                                       VARIANT* pData)
```

Descripción

Lee un bit en particular (Boolean), un byte (Byte), una palabra de 16 bits (Integer) o una palabra de 32 bits (Long) del área de las salidas periféricas (PQ).

Parámetros

ByteIndex	Representa la posición del byte inicial en el búfer de imagen de la periferia a leer. Los valores válidos de *ByteIndex* dependen de la CPU.
BitIndex	Representa la posición del bit (en bytes) en el búfer de imagen de la periferia a leer. Los valores válidos están comprendidos entre 0 y 7.
DataType	Una de las PointDataTypeConstants
pData	Puntero a los datos a leer. Los valores válidos de Data dependen del tipo de datos.

Notas

Si el parámetro *DataType* es un S7_Bit, *ByteIndex* y *BitIndex* deberán tener índices válidos. Si se ejecuta correctamente, el método devolverá el bit indicado en *pData* y su tipo de datos Variant será booleano.

Si el parámetro *DataType* es S7_Byte, S7_Word o S7_DoubleWord, *ByteIndex* deberá tener un índice válido (*BitIndex* se ignorará). Si se ejecuta correctamente, el método devolverá el valor en *pData*. El tipo de datos de Variant es Byte, Integer, o Long, dependiendo de *DataType*.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_BADBYTENDX	0x80040201 : Índice de byte no válido
PS_E_BADBYTECOUNT	0x80040202 : El tamaño del array de datos no es válido para el índice del byte inicial indicado
PS_E_READFAILED	0x80040203 : Ha fallado la operación de lectura
PS_E_BADBITNDX	0x80040205 : Índice de bit no válido
PS_E_BADTYPE	0x80040206 : Tipo de datos no válido
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado

Uso en Visual Basic

```
Function ReadOutputPoint(ByteIndex As Long, BitIndex As Long, DataType As PointDataTypeConstants, pData) As Long
```

SavePLC

STDMETHOD(**CS7ProSim::SavePLC**)(BSTR *FileName*)

Descripción

Guarda en un archivo los datos actuales del PLC simulado.

Los datos que se almacenan incluyen el programa, la configuración de hardware, la posición del selector de modo que indica la subventana "CPU", el tipo de ciclo (continuo o individual), el estado de E/S, los valores de los temporizadores, las direcciones simbólicas y el estado de la alimentación ("on" u "off").

Parámetros

FileName Nombre del archivo en el que deben guardarse los datos del PLC simulado

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
STG_E_CANTSAVE	0x80030103 : Imposible guardar
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Sub SavePLC(FileName As String)
```

SetScanMode

```
STDMETHOD(CS7ProSim::SetScanMode)( ScanModeConstants newVal)
```

Descripción

Ajusta el modo de ciclo de S7-PLCSIM.

Parámetros

newVal	Modo de ciclo a ajustar en S7-PLCSIM. El modo de ciclo debe ser una de las ScanModeConstants

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Sub SetScanMode(newVal As ScanModeConstants)
```

SetStartUpSwitch

```
STDMETHOD(CS7ProSim::SetStartUpSwitch)( RestartSwitchPosition postion)
```

Descripción

Ajusta el tipo de arranque (rearranque, rearranque completo o arranque en frío) de S7-PLCSIM.

Parámetros

postion Valor de la posición de arranque de S7-PLCSIM a ajustar

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Sub SetStartUpSwitch(postion As RestartSwitchPosition)
```

SetState

```
STDMETHOD(CS7ProSim::SetState)( BSTR newVal)
```

Descripción

Ajusta la posición actual del selector de modo de S7-PLCSIM (RUN, RUN-P o STOP).

Parámetros

newVal	Valor de la posición del selector de modo de S7-PLCSIM a ajustar

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
E_INVALID_STATE	0x00008002 : Estado no válido
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM

Uso en Visual Basic

```
Sub SetState(newVal As String)
```

StartPLCSim

`STDMETHOD(`**`CS7ProSim::StartPLCSim`**`)( BSTR` *`plcFile`*`)`

Descripción

Inicia S7-PLCSIM con el archivo indicado de simulación del PLC (guardado en una llamada anterior a SavePLC).

Parámetros

plcFile Nombre del archivo con el que debe iniciarse S7-PLCSIM

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado

Uso en Visual Basic

Sub **StartPLCSim**(*plcFile As String*)

WriteDataBlockValue

```
STDMETHOD(CS7ProSim::WriteDataBlockValue)(
                long BlockNumber,
                long ByteIndex,
                long BitIndex,
                const VARIANT* pData)
```

Descripción

Escribe un bit, un byte, una palabra o una palabra doble en el área de memoria DB de S7-PLCSIM.

Parámetros

BlockNumber	Representa el número del bloque de datos a escribir. Los valores válidos de *BlockNumber* dependen de la CPU.
ByteIndex	Representa la posición del byte inicial en el bloque de datos a escribir. Los valores válidos de *ByteIndex* dependen de la CPU.
BitIndex	Representa la posición del bit inicial en el bloque de datos a escribir (si se escribe un valor (de bit) booleano). Los valores válidos de *BitIndex* están comprendidos entre 0 y 7.
pData	Puntero al espacio que contiene los datos a escribir. Es preciso asignar y liberar esta área de memoria en la aplicación de usuario.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado
PS_E_BADTYPE	0x80040206 : Tipo de datos no válido
PS_E_BADBYTENDX	0x80040201 : Índice de byte no válido
PS_E_BADBYTECOUNT	0x80040202 : El tamaño del array de datos no es válido para el índice del byte inicial indicado
PS_E_WRITEFAILED	0x80040204 : Ha fallado la operación de escritura

Uso en Visual Basic

```
Sub WriteDataBlockValue(BlockNum As Long, ByteIndex As Long, BitIndex As Long, pData)
```

WriteFlagValue

```
STDMETHOD(CS7ProSim::WriteFlagValue)( long ByteIndex,
                                      long BitIndex,
                                      const VARIANT* pData)
```

Descripción

Escribe un bit, un byte, una palabra o una palabra doble en el área de memoria de marcas (M) de S7-PLCSIM.

Parámetros

ByteIndex	Representa la posición del byte inicial en la memoria M a escribir. Los valores válidos de *ByteIndex* dependen de la CPU.
BitIndex	Representa la posición del bit inicial en el byte de la memoria M a escribir (si se escribe un valor (de bit) booleano). Los valores válidos de *BitIndex* están comprendidos entre 0 y 7.
pData	Puntero al espacio que contiene los datos a escribir. Es preciso asignar y liberar esta área de memoria en la aplicación de usuario.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado
PS_E_BADTYPE	0x80040206 : Tipo de datos no válido
PS_E_BADBYTENDX	0x80040201 : Índice de byte no válido
PS_E_BADBYTECOUNT	0x80040202 : El tamaño del array de datos no es válido para el índice del byte inicial indicado
PS_E_WRITEFAILED	0x80040204 : Ha fallado la operación de escritura

Uso en Visual Basic

```
Sub WriteFlagValue(ByteIndex As Long, BitIndex As Long, pData)
```

WriteInputImage

```
STDMETHOD(CS7ProSim::WriteInputImage)( long StartIndex, const
VARIANT* pData)
```

Descripción

Escribe elementos en la imagen de las entradas de la periferia (área de memoria PI) de S7-PLCSIM, comenzando en el *StartIndex* de los datos a los que señala *pData*.

Parámetros

StartIndex	Representa la posición del byte inicial en el búfer de imagen de las entradas de periferia a escribir. Los valores válidos de *StartIndex* dependen de la CPU.
pData	Puntero a los datos de S7-PLCSIM a escribir. Los valores válidos de data dependen de la CPU. Es preciso asignar y liberar esta área de memoria en la aplicación de usuario.

Notas

El tipo de elementos a escribir es determinado por el tipo de elementos de Data. Todos los elementos deben pertenecer a un mismo tipo de datos. Un array Byte escribe bytes, un array Integer escribe palabras y un array Long escribe palabras dobles. Los valores escritos serán datos no procesados, por lo que el método no los interpretará ni los convertirá de manera alguna. El número de elementos escritos depende del tamaño del array al que señala Data.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_BADBYTENDX	0x80040201 : Índice de byte no válido
PS_E_BADBYTECOUNT	0x80040202 : El tamaño del array de datos no es válido para el índice del byte inicial indicado
PS_E_WRITEFAILED	0x80040204 : Ha fallado la operación de escritura
PS_E_BADTYPE	0x80040206 : Tipo de datos no válido
PS_E_NOTALLWRITESWORKED	0x80040210 : Han fallado todas las operaciones de escritura
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado

Uso en Visual Basic

```
Function WriteInputImage(StartIndex As Long, Data) As Long
```

WriteInputPoint

```
STDMETHOD(CS7ProSim::WriteInputPoint)( long ByteIndex,
                                       long BitIndex,
                                       const VARIANT* pData)
```

Descripción

Escribe un bit en particular (Boolean), un byte (Byte), una palabra de 16 bits (Integer) o una palabra de 32 bits (Long) del Data Variant en el área de las entradas de periferia (PI).

Parámetros

ByteIndex	Representa la posición del byte inicial en el búfer de imagen de las entradas de periferia a escribir. Los valores válidos de *ByteIndex* dependen de la CPU.
BitIndex	Representa la posición del bit (en bytes) en el búfer de imagen de la periferia a escribir. Los valores válidos están comprendidos entre 0 y 7.
pData	Puntero a los datos a escribir. Los valores válidos de Data dependen del tipo de datos.

Notas

Si el tipo de datos es booleano, *ByteIndex* y *BitIndex* deberán tener índices válidos. Si se ejecuta correctamente, el método escribirá el bit indicado en *pData*.

Si el tipo de datos indicado es Byte, Integer o Long, *ByteIndex* deberá tener un índice válido (*BitIndex* se ignorará). Si se ejecuta correctamente, el método escribirá los elementos en *pData*.

Tratamiento de errores

Los errores se devuelven en el evento ConnectionError y no en la llamada a la función.

Valor de retorno

Valor	Significado
S_OK	0x00000000 : Código de éxito
E_FAIL	0x80004005 : Error no especificado
PS_E_BADBYTENDX	0x80040201 : Índice de byte no válido
PS_E_BADBYTECOUNT	0x80040202 : El tamaño del array de datos no es válido para el índice del byte inicial indicado
PS_E_WRITEFAILED	0x80040204 : Ha fallado la operación de escritura
PS_E_BADBITNDX	0x80040205 : Índice de bit no válido
PS_E_BADTYPE	0x80040206 : Tipo de datos no válido
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado

Uso en Visual Basic

```
Function WriteInputPoint(ByteIndex As Long, BitIndex As Long, Data) As Long
```

Eventos

ConnectionError	Se genera si no es posible establecer una conexión con el control engine ("S7-PLCSIM") o cuando ocurre un error al llamar a un método de S7ProSim.
PauseStateChanged	Se genera al detectarse un cambio de estado de Pause/Continue. NewState es una cadena que representa una de las PauseStateConstants.
PLCSimStateChanged	Se genera al detectarse un nuevo estado del selector de modos de operación del PLC. NewState es el nuevo modo de operación: "RUN", "RUN_P" o "STOP".
ScanFinished	Se genera cuando se ejecuta un ciclo individual. ScanInfo proporciona informaciones indexadas acerca del ciclo.
ScanModeChanged	Se genera cuando se detecta un cambio de ScanMode. NewState es una cadena que representa una de las ScanModeConstants.

ConnectionError

HRESULT **ConnectionError**(BSTR *ControlEngine*, long *Error*)

Descripción

Se genera si no es posible establecer una conexión con el control engine ("S7-PLCSIM") o cuando ocurre un error al llamar a un método de S7ProSim.

Uso en Visual Basic

Event **ConnectionError**(*ControlEngine As String*, *Error As Long*)

PauseStateChanged

```
HRESULT PauseStateChanged(BSTR NewState)
```

Descripción

Se genera al detectarse un cambio de estado de Pause/Continue. NewState es una cadena que representa una de las PauseStateConstants.

Uso en Visual Basic

```
Event PauseStateChanged(NewState As String)
```

PLCSimStateChanged

```
HRESULT PLCSimStateChanged(BSTR NewState)
```

Descripción

Se genera al detectarse un nuevo estado del selector de modos de operación del PLC. NewState es el nuevo modo de operación: "RUN", "RUN_P" o "STOP".

Uso en Visual Basic

```
Event PLCSimStateChanged(NewState As String)
```

ScanFinished

```
HRESULT ScanFinished(VARIANT ScanInfo)
```

Descripción

Se genera cuando se ejecuta un ciclo individual. ScanInfo proporciona informaciones indexadas acerca del ciclo.

Uso en Visual Basic

```
Event ScanFinished(ScanInfo)
```

ScanModeChanged

```
HRESULT ScanModeChanged(BSTR NewState)
```

Descripción

Se genera cuando se detecta un cambio de ScanMode. NewState es una cadena que representa una de las ScanModeConstants.

Uso en Visual Basic

```
Event ScanModeChanged(NewState As String)
```

Definiciones de tipo

CPURunMode	Constantes del estado de ciclo de la CPU en modo RUN
ImageDataTypeConstants	Constantes del método ReadOutputImage
PauseStateConstants	Constantes del estado de pausa
PointDataTypeConstants	Constantes del método ReadOutputPoint
RestartSwitchPosition	Constantes de la posición del interruptor de arranque en el panel frontal
ScanModeConstants	Constantes del modo de ciclo
tagPauseState	Constantes del estado de pausa
Constantes ScanInfo	Constantes de información acerca del ciclo

CPURunMode

```
enum CPURunMode { CONTINUOUS_SCAN, SINGLE_SCAN, SINGLE_STEP }
```

Descripción

Constantes del estado de ciclo de la CPU en modo RUN

Miembros

CONTINUOUS_SCAN
SINGLE_SCAN
SINGLE_STEP

ImageDataTypeConstants

```
enum {
        S7Byte = 2,
        S7Word = 3,
        S7DoubleWord = 4
    }
```

Descripción

Constantes del método ReadOutputImage

Miembros

S7Byte
S7DoubleWord
S7Word

PauseStateConstants

```
enum {
        Running = 0,
        Paused = 1,
        Disabled = 2
    }
```

Descripción

Constantes del estado de pausa

Miembros

Disabled

Paused

Running

PointDataTypeConstants

```
enum {
        S7_Bit = 1,
        S7_Byte = 2,
        S7_Word = 3,
        S7_DoubleWord = 4
    }
```

Descripción

Constantes del método ReadOutputPoint

Miembros

S7_Bit
S7_Byte
S7_DoubleWord
S7_Word

RestartSwitchPosition

```
enum {
        WarmStart = 0,
        HotStart = 1,
        ColdStart = 2
    }
```

Descripción

Constantes de la posición del interruptor de arranque en el panel frontal

Miembros

ColdStart	Posición de rearranque OB102
HotStart	Posición de rearranque OB101
WarmStart	Posición de rearranque OB100

ScanModeConstants

```
enum {
        SingleScan = 0,
        ContinuousScan = 1
    }
```

Descripción

Constantes del modo de ciclo

Miembros

ContinuousScan
SingleScan

tagPauseState

```
enum tagPauseState { ENABLED_RUNNING, ENABLED_PAUSED,  DISABLED }
```

Descripción

Constantes del estado de pausa

Miembros

DISABLED
ENABLED_PAUSED
ENABLED_RUNNING

ScanInfo Constants

Constantes ScanInfo

ScanInfo	
NUM_OF_SCANINFO_ELEMENTS	Número de elementos en el array de retorno de ScanInfo.
EXECUTION_TIME_NDX	Índice 0: tiempo de ejecución (en ms)
MIN_CYCLE_TIME_NDX	Índice 1: tiempo de ejecución mínimo (en ms)
LARGEST_CYCLE_TIME_NDX	Índice 2: tiempo de ejecución máximo (en ms)
AVERAGE_CYCLE_TIME_NDX	Índice 3: tiempo de ciclo promedio (en ms)
IS_PLC_RUNNING_NDX	Índice 4: marca: 1=PLC está funcionando; 0=PLC no está funcionando

ScanInfo

ScanInfo

El tipo de datos **ScanInfo** representa un array de longs. Cada long en el array define algunas informaciones acerca del ciclo, conforme con las constantes ScanInfo.

NUM_OF_SCANINFO_ELEMENTS

```
#define NUM_OF_SCANINFO_ELEMENTS    5
```

Descripción

Número de elementos en el array de retorno de ScanInfo.

EXECUTION_TIME_NDX

```
#define EXECUTION_TIME_NDX          0
```

Descripción

Índice 0: tiempo de ejecución (en ms)

MIN_CYCLE_TIME_NDX

```
#define MIN_CYCLE_TIME_NDX          1
```

Descripción

Índice 1: tiempo de ejecución mínimo (en ms)

LARGEST_CYCLE_TIME_NDX

```
#define LARGEST_CYCLE_TIME_NDX       2
```

Descripción

Índice 2: tiempo de ejecución máximo (en ms)

AVERAGE_CYCLE_TIME_NDX

```
#define AVERAGE_CYCLE_TIME_NDX       3
```

Descripción

Índice 3: tiempo de ciclo promedio (en ms)

IS_PLC_RUNNING_NDX

```
#define IS_PLC_RUNNING_NDX           4
```

Descripción

Índice 4: marca: 1=PLC está funcionando; 0=PLC no está funcionando

Códigos de error

PS_E_BADBITNDX	0x80040205 : Índice de bit no válido
PS_E_BADBYTECOUNT	0x80040202 : El tamaño del array de datos no es válido para el índice del byte inicial indicado
PS_E_BADBYTENDX	0x80040201 : Índice de byte no válido
PS_E_BADTYPE	0x80040206 : Tipo de datos no válido
PS_E_INVALIDCALLBACK	0x80040207 : Devolución de llamada no válida
PS_E_INVALIDDISPATCH	0x80040208 : Distribución no válida
PS_E_INVALIDINPUT	0x80040213 : Introducción no válida
PS_E_INVALIDSCANTYPE	0x8004020B : Tipo de de ciclo no válido, debe ser una de las ScanModeConstants
PS_E_MODENOTPOSSIBLE	0x8004020C : S7-PLCSIM no ha podido activar el modo de ciclo indicado
PS_E_NOTALLREADSWORKED	0x8004020F : Han fallado todas las operaciones de lectura
PS_E_NOTALLWRITESWORKED	0x80040210 : Han fallado todas las operaciones de escritura
PS_E_NOTCONNECTED	0x80040211 : S7ProSim no está conectado a S7-PLCSIM
PS_E_NOTIFICATION_EXIST	0x8004020D : S7ProSim ya está registrado para la notificación
PS_E_NOTREGISTERED	0x80040209 : S7ProSim no está registrado para devoluciones de llamadas de S7-PLCSIM
PS_E_NOTSINGLESCAN	0x8004020A : S7-PLCSIM no está en modo de ciclo individual
PS_E_PLCNOTRUNNING	0x8004020E : S7-PLCSIM is not running
PS_E_POWEROFF	0x80040212 : S7-PLCSIM está desconectado
PS_E_READFAILED	0x80040203 : Ha fallado la operación de lectura
PS_E_WRITEFAILED	0x80040204 : Ha fallado la operación de escritura
E_FAIL	0x80004005 : Error no especificado
E_INVALID_STATE	0x00008002 : Estado no válido
S_OK	0x00000000 : Código de éxito
STG_E_CANTSAVE	0x80030103 : Imposible guardar

Índice

A

AVERAGE_CYCLE_TIME_NDX, 55, 57

B

BeginScanNotify, 13

C

Cargar archivo .plc guardado, 35

Cargar E/S de periferia durante el rearranque, 25

Ciclo

- Ajustar, 32
- Obtener, 22

Códigos de error, 59

ColdStart, 52

- Ajustar, 33

Connect, 14

ConnectionError, 42

Constantes definidas, 47

- CPURunMode, 48
- ImageDataTypeConstants, 49
- PauseStateConstants, 50
- PointDataTypeConstants, 51
- RestartSwitchPosition, 52
- ScanModeConstants, 53
- tagPauseState, 54

Constantes ScanInfo, 55

Continue, 15

Controladores de eventos, 8

CPURunMode, 48

D

Definiciones de tipo, 47

- CPURunMode, 48
- ImageDataTypeConstants, 49
- PauseStateConstants, 50
- PointDataTypeConstants, 51
- RestartSwitchPosition, 52
- ScanModeConstants, 53
- tagPauseState, 54

Disconnect, 16

E

Ejecutar ciclo continuo

- GetScanMode, 22
- ScanModeConstants, 53
- SetScanMode, 32

Ejecutar ciclo individual

- ExecuteNmsScan, 18
- ExecuteNScans, 19
- ExecuteSingleScan, 20
- GetScanMode, 22
- ScanModeConstants, 53
- SetScanMode, 32

EndScanNotify, 17

Escribir

- Entrada, 39
- Imagen de las entradas, 38
- Valores de marcas (memoria M), 37
- Valores del bloque de datos (DB), 36

Estado de pausa

- Ajustar, 15, 26
- Obtener, 21

Eventos, 41

- ConnectionError, 42
- PauseStateChanged, 43
- PLCSimStateChanged, 44
- ScanFinished, 45
- ScanModeChanged, 46

ExecuteNmsScan, 18

ExecuteNScans, 19

ExecuteSingleScan, 20

EXECUTION_TIME_NDX, 55, 56

G

GetPauseState, 21

GetScanMode, 22

GetStartUpSwitch, 23

GetState, 24

Guardar archivo .plc, 31

H

HotStart, 52
  Ajustar, 25, 33
HotStartWithSavedValues, 25

I

ImageDataTypeConstants, 49
Introducción, 7
IS_PLC_RUNNING_NDX, 55, 57

L

LARGEST_CYCLE_TIME_NDX, 55, 57
Leer
  Imagen de las salidas, 29
  Salida, 30
  Valores de marcas (memoria M), 28
  Valores del bloque de datos (DB), 27

M

Marcas
  Escribir, 37
  Leer, 28
Memoria de marcas (M)
  Escribir, 37
  Leer, 28
Métodos, 11
  BeginScanNotify, 13
  Connect, 14
  Continue, 15
  Disconnect, 16
  EndScanNotify, 17
  ExecuteNmsScan, 18
  ExecuteNScans, 19
  ExecuteSingleScan, 20
  GetPauseState, 21
  GetScanMode, 22
  GetStartUpSwitch, 23
  GetState, 24
  HotStartWithSavedValues, 25
  Pause, 26
  ReadDataBlockValue, 27
  ReadFlagValue, 28
  ReadOutputImage, 29
  ReadOutputPoint, 30
  SavePLC, 31
  SetScanMode, 32
  SetStartUpSwitch, 33
  SetState, 34
  StartPLCSim, 35
  WriteDataBlockValue, 36
  WriteFlagValue, 37
  WriteInputImage, 38
  WriteInputPoint, 39
Métodos de ejecución de ciclos
  ExecuteNmsScan, 18
  ExecuteNScans, 19
  ExecuteSingleScan, 20
MIN_CYCLE_TIME_NDX, 55, 56

N

Notificación de ciclo, 13, 17
NUM_OF_SCANINFO_ELEMENTS, 55, 56

O

Objeto COM S7ProSim de Siemens, agregar al proyecto, 7

P

Pause, 26
PauseStateConstants, 50
PauseStateChanged, 43
PLCSIM, iniciar, 35
PLCSimStateChanged, 44
PointDataTypeConstants, 51
Posición del interruptor de arranque, 52
  Ajustar, 33
  Obtener, 23
Posición del selector de modo
  Ajustar, 34
  Obtener, 24
Programar una interfaz de S7ProSim con S7-PLCSIM, 8
Proyecto Visual Basic, agregar S7ProSim, 7

R

ReadDataBlockValue, 27
ReadFlagValue, 28

ReadOutputImage, 29
ReadOutputPoint, 30
Referencias, 7
Referencias del proyecto, 7
RestartSwitchPosition, 52
Resumen breve, 7

S

S7-PLCSIM, iniciar, 25, 35
S7ProSim
  Agregar a un proyecto VB, 7
  Interfaz con S7-PLCSIM, programar, 8
  Resumen breve, 7
SavePLC, 31
ScanFinished, 45
ScanModeConstants, 53
ScanModeChanged, 46
SetScanMode, 32
SetStartUpSwitch, 33
SetState, 34
StartPLCSim, 35

T

tagPauseState, 54
Tipos enumerados, 47
  CPURunMode, 48
  ImageDataTypeConstants, 49
  PauseStateConstants, 50
  PointDataTypeConstants, 51
  RestartSwitchPosition, 52
  ScanModeConstants, 53
  tagPauseState, 54

V

Valores de retorno, 59
Valores del bloque de datos (DB)
  Escribir, 36
  Leer, 27

W

WarmStart, 52
  Ajustar, 33
WriteDataBlockValue, 36
WriteFlagValue, 37
WriteInputImage, 38
WriteInputPoint, 39